Charge Redistribution ADC

Fuding Ge

Autozeroing

In most applications, the comparator does not operate continuously, but rather makes a comparison between two voltages and then is reset. During the reset phase, it is possible to apply autozero technique. This technique works well with CMOS because of the high input resistance of the CMOS devices.

Problem with the autozeroing: When the CMOS switches open and close, **charge is injected or removed** by the large clock swings on the gate of the switches. Second the capacitor C_{az} will lose some of its charge during the cancellation phase. Third, if the offset is completely cancelled for a given common mode value of Vcom and Vin during Pha1, it may not cancel during Pha2 because a different common mode value of Vcom and Vin may have a difference value of offset voltage. There is also the problem of stability because the comparator has unit gain, negative feedback applied to it during the sampling mode. It is necessary to compensate the comparator. Finally, a noise KT/Caz is injected into the circuit each time the switch closes.

SAR

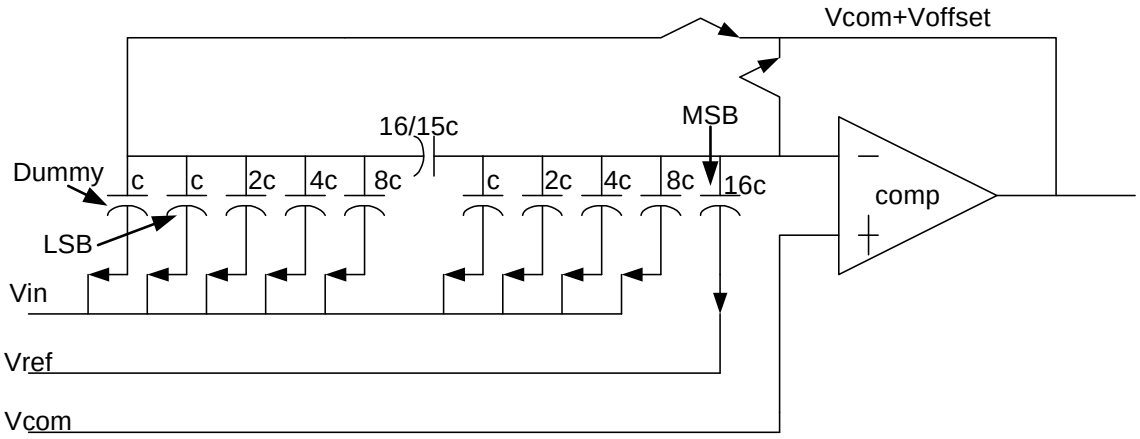
The function of the SAR is determining each bit in a sequential manner, based on the output of the comparator. If an N bit A/D successive approximation converter is implemented, there are 2^N possible conversion output values so the SAR needs at least 2^N States and so N FF. A non-redundant SAR with the minimum FF was used in this design.

At the beginning of the conversion (initialization step) the SAR assumes that the MSB is 1 and all other bits are 0.

Conversion Step	Input D/A word	Comparator output
0	1 0 0 0 0 0 0 0 0	a8
1	a8 1 0 0 0 0 0 0 0	a7
2	a8 a7 1 0 0 0 0 0 0	a6
3	a8 a7 a6 1 0 0 0 0 0	a5
4	a8 a7 a6 a5 1 0 0 0 0	a4
5	a8 a7 a6 a5 a4 1 0 0 0	a3
6	a8 a7 a6 a5 a4 a3 1 0 0	a2
7	a8 a7 a6 a5 a4 a3 a2 1 0	a1
8	a8 a7 a6 a5 a4 a3 a2 a1 1	a0
Result	a8 a7 a6 a5 a4 a3 a2 a1 a0	-

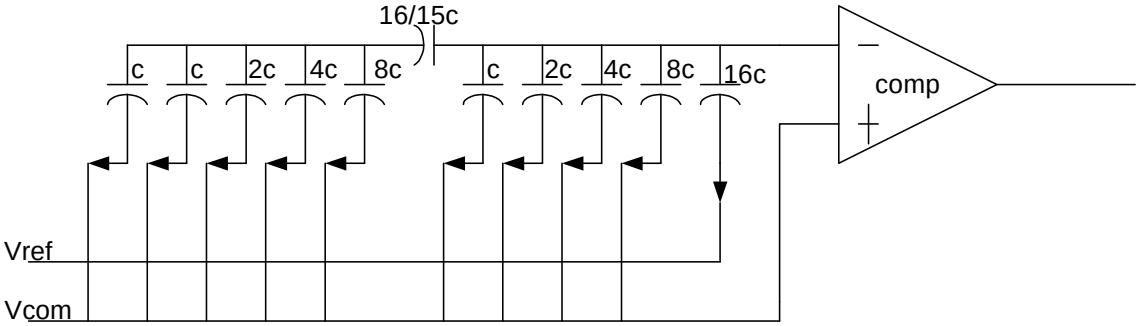
a8 is MSB in the above table.
The successive approximation sequence can be described by the following algorithm.

1. **Sampling:**

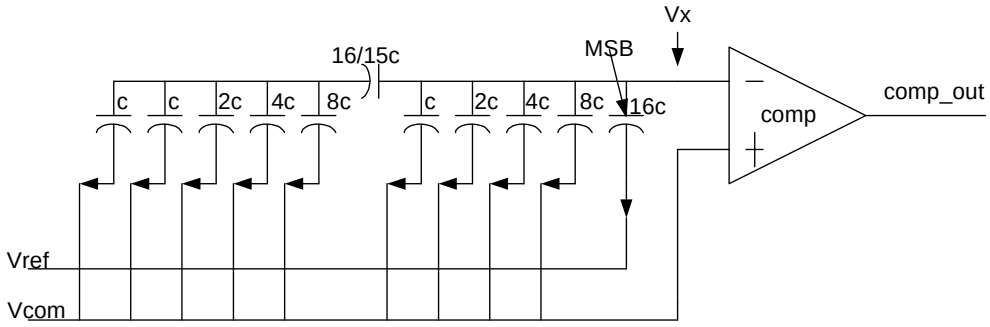


Autozeroing offset cancellation is accomplished in this step.

2. Hold mode

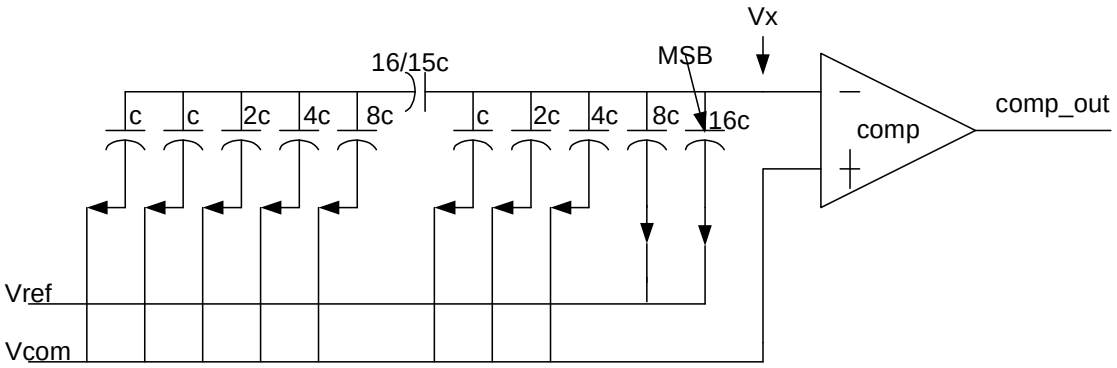


3. MSB Bit Test Mode



If Vx > Vcom, --> Vin < Vcom, --> bottom plate of MSB switch to vcom for next bit test,--> level shift
If Vx < Vcom, --> Vin > Vcom, --> bottom plate of MSB continue to connect to Vref for next bit test,--> no level shift

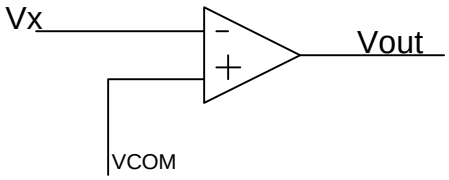
3. Other Bits Test Mode



If Vx > Vcom, --> bottom plate of the test bit switch to vcom for next bit test
If Vx < Vcom, --> bottom plate of the test bit continue to connect to Vref for next bit test
After LSB is test, Vx is equal to or very close to Vcom

At the end of test mode (LSB is determined), Vx à Vcom.

Configuration of test:



At the beginning of the conversion (initialization step) the SAR assumes that the MSB is 1 and all other bits are 0. This digital word is applied to the D/A converter, which generates an analog signal of 0.5 FS (FS is the range for the analog input Vin), result in Vx=-Vin+Vref/2, Vx is then compared to Vcom. If the comparator output is high, means Vx<Vcom, the digital control makes the MSB 1. This makes MSB capacitor continue to connect to Vcom. If the comparator output is low, the digital control makes the MSB 0. At this point the value of MSB is known. The successive approximation steps are performed by once more applying a digital word to the D/A converter, with MSB has its proven value, the second bit guessed at 1 and all the remaining bits having a value of 0. Again the sampled input is compared to the output of the D/A converter with this digital word applied: if the output of the comparator is high, the second bit is proven to be 1 else 0 and so on until the LSB is determined.

For the above function description we define the SAR as a sequential FSM. Consider the m conversion step for the (m+1) step the generic bit k^{th} (k=0, 1, 2, 3, …, N) can take the value of:

- The $(k+1)^{th}$ bit on the left more significant than k^{th} (k-1, k-2, …, 0) if all the less significant bits on the right anf the k^{th} bit itself have value of 0.
- The output of the comparator if all the less significant bits than the k^{th} (k-1, k-2, …, 0) have value of 0 and the k^{th} itself has value 1.
- The k^{th} bit at the (m-1) step if the at least one of the less significant bit than the k^{th} (k-1, k-2, …, 0) is 1.

The basic structure of the SAR is a multiple input N bit shift register. To start the conversion the MSB FF and all the other FFs are forced to the initialization state. It is necessary to provide a mechanism that brings the shift register in the initialization state: using set and reset inputs connect to the control signal (START) which, if active, sets the MSB FF and reset the all other bits. For the next states, a generic FF (k^{th}) must have the possibility of choosing between three data inputs coming from:

- The output of the $(k+1)^{th}$ FF (shift right)
- The output of the comparator (a) (data load)
- The output of the k^{th} FF itself (memorization).

By adding the a multiplexer and a decoder to each FF the three inputs can be selected. The all-zero state for the less significant bits than the k^{th} (k-1, k-2, …, 0) is revealed with an OR-chain of the outputs of the FF storing them. The decoding logic for the k^{th} multiplexer needs these two selecting inputs to put the SAR in the correct operation mode: the output of the OR-chain of the precedent FF (or_out) and the out put of the k^{th} FF itself (qk).

or-our	qk	Operation
1	X	Memorization (qk)
0	1	Data load (comp_out)
0	0	Shift right (qk_pre)

To stop the conversion, we can use the OR chain applying to the LSB a control signal (STOP) which, if active, force a “1” in all signals A so that the SAR operates in the memorization mode. The STOP signal is useful in particular when the successive approximation sequence reaches the last step (end of conversion) and the conversion can stored in SAR. The output of the last FF can be used to signal the end of conversion.

This design used only one single voltage reference Vref to implement bipolar voltage inputs refers to VCOM=1.5V. This is achieved by connecting the bottom plate of the largest capacitor to Vref during the sample mode resulting in a stored charge:

$$Q_x=-Ctot(Vin/2 +Vref/2)$$

For unipolar case, Q_x=-Ctot*Vin.
Each bit is then tested in sequence just as the unipolar case except that the largest capacitor is switched from Vref to Vcom during its test while all the other capacitors are switched from Vcom to Vref. Also as unipolar case, a bit value is true if Vx is smaller than VCOM. The expression converge back to VCOM as in the unipolar case:

Capacitor Array

Unit capacitor consideration:

Smallest capacitor size estimation:

For 8-bit resolution Δ=1LSB=5.5mV.

For 9-bit resolution Δ=1LSB=2.75mV

Quantization noise is $N_q^2 = 1LSB^2/12$

For 8-bit resolution: $N_q^2 = 2.62\ mV^2$.

For 9-bit resolution: $N_q^2 = 2.62\ mV^2$.

$Nt=\sqrt{(2.62+KT/C+N_{thermal})} < 0.5LSB$. Neglect the thermal noise,

$K=1.38*10^{-23}$, T=300. For 1pF, $=\sqrt{(KT/C)}=64uV$.
What we have is
$\sqrt{(KT/1pF)}=64uV$
$\sqrt{(KT/CpF)}=0.5LSB$

$C\ (pF)=(64uV/0.5LSB(uV))^2$.

In this design,
For 8-bit resolution: 0.5LSB=2.8mV, **$C=(64/2800)^2=0.5fF$**.
For 9-bit resolution: 0.5LSB=1.4mV, **$C=(64/1400)^2=2.1fF$**.

Nonlinearity of Capacitor array DACSs arises from three sources, capacitor mismatch, capacitor nolinearity and the nonlinearity of the junction capacitance of any switches connected to the output node. Mismatch arises from both gradients and random variations.

$$\Delta C/C = \Delta W/W + \Delta L/L - \Delta tox/tox$$

This expression indicates that increasing W, L or tox improving matching. In practice, tox is a constant of the process, leaving W and L as the only variables. However, increase the W and L does not reduce ΔC/C indefinitely because tox gradients become more significant at large dimension. As a consequences, ΔC/C reaches a minimum at certain optimum dimensions.

Accuracy:

$$|INL|_{max} = 2^{N-1}|\Delta C|_{max,\ INL}$$

where the maximum ΔC that will result in an INL is less than ½ LSB is

$$|\Delta C|_{max,\ INL} = 0.5C/^{2N-1} = C/2^N$$

DNL is defined by $DNL_{max} = (2^N-1) \bullet |\Delta C|_{max, DNL}$
With the maximum ΔC which will result in a DNL less than ½ LSB is

$$|\Delta C|_{max, DNL} = 0.5C/2^N-1 = C/(2^{N+1}-2)$$

Capacitor Array Switches Decoder

3 inputs: Vref, Vcom, Vin
3 Control signals: Start (when 1 à Vin); Stop (when 1 à Vcom); bitk (when 1à Vref)

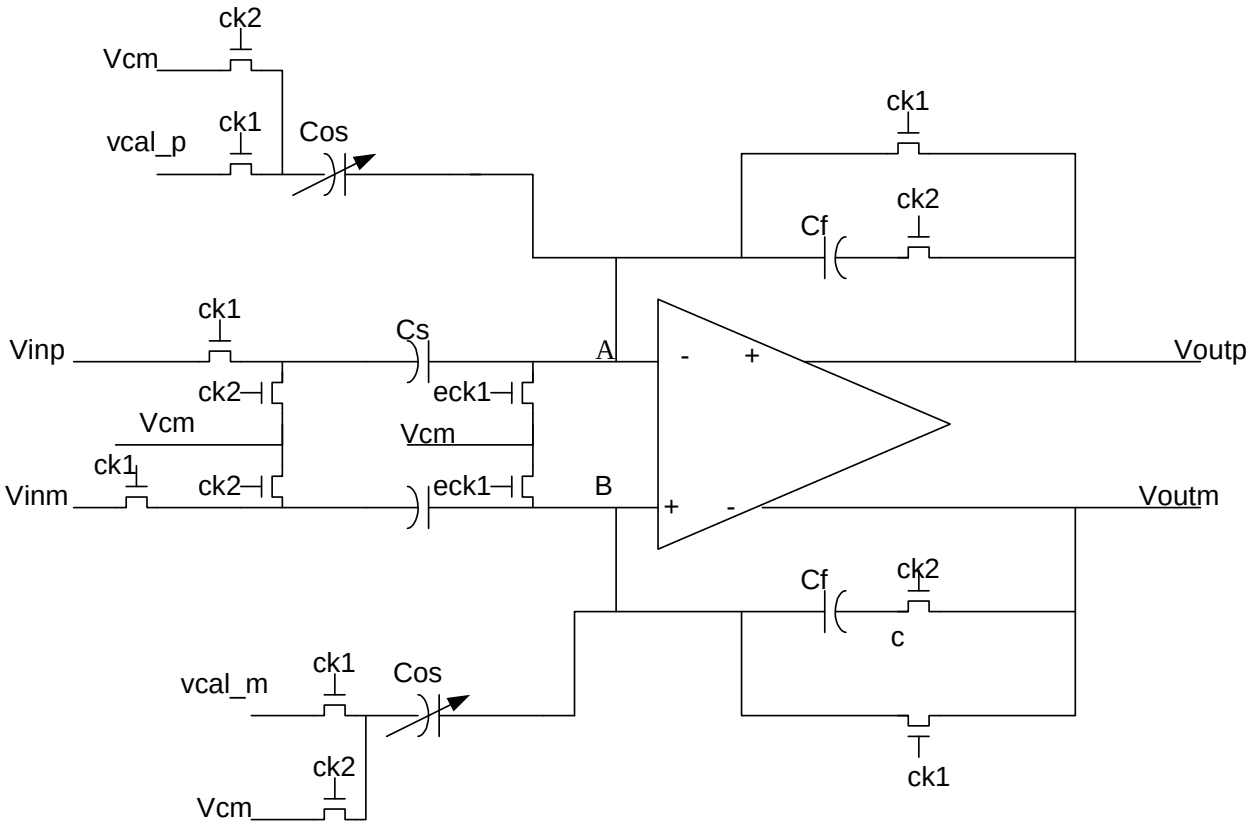
Start	Stop	q_k	Pass signal
1	**x**	**x**	Vin
x	**1**	**x**	Vcom
0	**0**	**1**	Vref
0	**x**	**0**	Vcom

For level shift, the **LSB (bit8)**
decoder is different, it is

Start bit8	Stop		Pass signal	Note
1	**x**	**x**	Vref	
x	**1**	**x**	Vref	
0	**0**	**1**	Vcom	Vin<Vcom (1.5), level shift, Bit8=1 means Vin < Vcom
0	**x**	**0**	Vref	Vin>Vcom, no level shift

Offset Calibration

The basic principle is shown in the following figure. The calibrated offset value can bee added or subtracted from voutp and voutm.



When ck1 is high, Cos (programmble capacitor array) is charged to

$$Q=Cos*(Vcm-Vcml)=Cos*\Delta V$$

Where ΔV = Vcm- vcal_p or Vcm- vcal_m.

When ck2 is high, nodes A and B are in virtual Vcm. So the voltage at both plates of Cos is Vcm and the charge Q= Cos*ΔV should redistribute to Cf. Then the output voltage voutp or voutm has a change due to Cos, which is:

$$Vos=-Q/Cf=-Cos/Cf*\Delta V$$